DUPLICATING PROCESSORS AND METHOD FOR CONTROLLING ANOMALOUS DUAL STATE THEREOF

BACKGROUND OF THE INVENTION

1. Field of the Invention

The present invention relates to a communication system, and more particularly, to duplicating processors and a method for controlling anomalous dual state of the duplicated processors.

2. Description of the Background Art

Generally, in order to improve reliability and stability in service of a communication system, a hardware path that transmits data is implemented by duplicating(active mode/standby mode), of which an active path is set at an initial stage through which data is transmitted, and if a disturbance occurs in the active path, a separately provided standby path is automatically switched to successively operated.

However, a communication system having the tightly coupled active/standby scheme that is physically constructed by hardware has disadvantages. The hardware architecture should be redesigned and a new .operating system on it should be required. In addition, expense and time are required too much to develop a new programming language.

In an effort to overcome the drawbacks, as shown in Figure 1, recent communication systems are established in that two processors 10 and 20 are loosely coupled through a network and heartbeat signals (HB_Tx/HB_Rx) that are periodically transmitted and received between the processors are used to process the duplication state by software.

For this purpose, the processors 10 and 20 respectively include blocks for performing the duplication function.

With reference to Figure 2, the processor A 10 includes an incoming heartbeat processing block 11, a duplication FSM(Finite State Machine) processing block 12 and an outgoing heartbeat processing block 13. The processor B 20 has the same flow.

The incoming heartbeat processing block 11 receives a heartbeat (HB_Rx) from the processor B 20, that is, the other processor (twin) and transfers state information of the twin 20 to the duplication FSM processing block 12. If no heartbeat is received from the twin 20 within a predetermined time, it reports a network disturbance or a twin down to the duplication FSM processing block 12.

The duplication FSM processing block 12 is charged with a corresponding state transition function according to the state information of the twin 20 included in the heartbeat (HB_Rx) or a switching event captured by the incoming heartbeat processing block 11, so that it applies the state information of itself corresponding to each state to the outgoing heartbeat processing block 13 or renders the outgoing heartbeat processing block 13 to transmit a heartbeat signal immediately in every state transition.

The outgoing heartbeat processing block 13 transmits the heartbeat (HB_Tx) to the twin 20 immediately or periodically according to the state information applied from the duplication FSM processing block 12.

Figure 3 illustrates a state transition of the duplication FSM block in accordance with a conventional art.

The state transition process in accordance with the conventional art will now be described with reference to Figure 3.

Each state transition is made by a twin state event such as 'Twin START', 'Twin ACTIVE' or 'Twin TIMEOUT' or by an external event such as 'Shutdown Command', 'Restart' or 'Manual Switchover'.

First, as the FSM is driven and all blocks on overall system are completely initialized, the duplication FSM allows the "INITIAL" state to transit to the "START" state.

And, the self processor confirms a state of the twin, and If the twin has been also started, the self processor is transited to 'NEGOTIATION' state to determine which side gives services as an active one. In the 'NEGOTIATION' state, it was predetermined that which of either one of two processors is to be active.

For example, if the processor A is set as an active processor, each processor confirms whether itself is the processor A in the 'NEGOTIATION' state. If either processor confirms itself as the processor A, it is transited to 'ACTIVE' state, or otherwise, it is transited to 'STANDBY' state.

Meanwhile, when the processor A is in 'ACTIVE' state, if the twin is in 'ACTIVE' state or if 'Manual switchover' occurs, the processor A is transited to 'STANDBY' state. And, if a network error or a disturbance occurs, the processor A is transited to 'PENDING STANDBY' state.

When the processor A is in 'PENDING STANDBY' state, it confirms a state of twin, and if the twin that is, the processor B is in 'ACTIVE' state, the processor A is transited to a 'SYNCH' state and then transited to 'STANDBY' state when synchronization is completed, while if the processor B is in 'STANDBY' state, the processor A is transited to ‘ACTIVE’ state.

Meanwhile, in case that the ‘Manual switchover’ occurs or the processor B is ‘Timeout’ before synchronization is completed, the processor A is transited to ‘ACTIVE’ state.

When the processor A is in ‘STANDBY’ state, if ‘Manual switchover’ occurs, the processor A is transited to ‘ACTIVE’ state. In case that the twin(processor B) is in ‘STANDBY’ state, the processor A is transited to ‘PENDING ACTIVE’ state and confirms a state of the twin. If the processor B is in ‘STANDBY’ state, the processor A is transited to ‘ACTIVE’ state, or otherwise, it is transited to ‘STANDBY’ state.

However, in case that the duplication is implemented by software through network as described above, there is a possibility that the network disturbance occurs or the network resources such as a cable or a hub are defected or disturbed. Then, each processor would judge that the twin has gone down, resulting in that both processors become active, which makes a confusion for external network elements/participants that mutually works with the processors, causing a problem failing to perform a normal operation.

In addition, even though the disturbance is restored, at just the time of restoration of the disturbance, it may fall to an anomalous dual state, though it happens at few random.

Moreover, if the two processors become all ‘ACTIVE’ state, since they recognize the other party as being in ‘ACTIVE’ state based on the received heartbeat, itself is accordingly transited to the ‘STANDBY’ state immediately. Meanwhile, in case that the two processors become all in ‘STANDBY’ state, since they recognize the other party as being in ‘STANDBY’ state based on the received heartbeat, itself is accordingly transited to the 'PENDING ACTIVE' state immediately. The difficulty is met in case that the twin may do the same action at the time.

If the twin is not the 'PENDING ACTIVE' nor 'ACTIVE', itself is transited to the 'ACTIVE' state. In this respect, normally, there is time differences to an extent in receiving the heartbeat, so that it may be prevented from falling into a double active state out of the 'PENDING ACTIVE' state.

That is, at this stage, differences are made in the receiving intervals of the heartbeat, so that the party that first reaches the 'PENING ACTIVE' state is transited to an active state and the party that reaches later is transited to the 'STANDBY' state, thereby maintaining a normal state.

However, if the heartbeat is transmitted or received at the accurately same time, an anomalous dual active/standby state is inevitably caused. Then, a state fluctuation phenomenon may occur that transition is made to the dual active/standby state, failing to perform a normal duplication.

SUMMARY OF THE INVENTION

Therefore, an object of the present invention is to provide duplicating processors and a method for controlling anomalous dual state in which seeds for generating random numbers are differently allocated when each processor is initialized to generate the different random number and transmission period of a heartbeat is continuously changed by using the random number, thereby avoiding an anomalous dual state.

Another object of the present invention is to provide a method for controlling duplicating processors which is capable of quickly restoring an anomalous dual state even though it occurs due to an abnormally on a network or on a system.

To achieve these and other advantages and in accordance with the purpose of the present invention, as embodied and broadly described herein, there is provided a method for controlling anomalous dual state of duplicated processors for a duplication system having a first and a second processors that are connected to each other through network, including the steps of: transmitting its own state information of either the first or the second processor to mutually another processor (twin) by using different transmission period to each other; receiving the heartbeat applied from the other processor and recognizing state information of the twin; and performing duplication states according to the state information of the twin.

In order to achieve the above objects, there is also provided duplicating processors in a fault-tolerant system having a first and a second processors that are mutually connected through a network, of which each processor has an outgoing heartbeat processing block for transmitting a heartbeat including its own state information to the other processor (twin) by using a different period to each other; an incoming heartbeat processing block for receiving the heartbeat from the other processor and recognizing the state information of the twin; and a duplication FSM processing block for performing duplication states processing according to the state information of the twin.

BRIEF DESCRIPTION OF THE DRAWINGS

The accompanying drawings, which are included to provide a further understanding of the invention and are incorporated in and constitute a part of this specification, illustrate embodiments of the invention and together with the description serve to explain the principles of the invention.

In the drawings:

Figure 1 is a block diagram illustrating a simple duplication architecture of a system through network in accordance with a conventional art and the present invention;

Figure 2 is a block diagram illustrating blocks for performing duplication process in each processor in accordance with the conventional art and the present invention;

Figure 3 illustrates a duplication FSM diagram in a duplication FSM processing block in accordance with the conventional art and the present invention;

Figure 4 is a flow chart of a process for transmitting a heartbeat of a outgoing heartbeat processing block in accordance with the present invention; and

Figure 5 is a flow chart of a process for receiving heartbeat of an incoming heartbeat processing block in accordance with the present invention.

DETAILED DESCRIPTION OF THE PREFERRED EMBODIMENTS

Reference will now be made in detail to the preferred embodiments of the present invention, examples of which are illustrated in the accompanying drawings.

The construction of a basic duplicated system for executing the present invention is the same as those of Figures 1 and 2 and its duplication state transition process is the same as that of Figure 3, for which, thus, descriptions are omitted.

Figure 4 is a flow chart of a process for transmitting a heartbeat of an outgoing heartbeat processing block in accordance with the present invention; and Figure 5 is a flow chart of a process for receiving heartbeat of an incoming heartbeat processing block in accordance with the present invention.

The process for controlling duplication state of a processor in accordance with the present invention will now be described with reference to the accompanying drawings.

The duplicated processors A and B 10 and 20 use heartbeats (HB_Tx/HB_Rx) that are mutually transmitted and received therebetween to inform the twin of its own state information and monitor the state of the twin. In this respect, in order to produce a continuous difference for the heartbeat transmission intervals, random number periods are generated.

For this purpose, first, when each processor 10 and 20 is initialized, seeds for different random numbers are allocated to generate random numbers, and the time tuned by a generated random number is used as a transmission period of the heartbeat of the outgoing heartbeat processing block 13.

In order to generate a suitable tuned period, an average transmission time 'a' through a link between the processors A and B 10 and 20, an average heartbeat processing time of processors 'b' and a state transition time 'c' should be considered.

Before each processor receives nth heartbeat from the twin, each processor should be already completed processing of the n-1th heartbeat and only one heartbeat message at its maximum should exist in the transmission path of a corresponding heartbeat at the point of a specific time.

Accordingly, a fixed-period heartbeat transmission period 'x' should satisfy the following formula: $0 < (2a+b+c) < x$. And, assuming that a heartbeat transmission period to be changed is 'p', the variable period and a maximum tolerance of a period change, that is, $|p-x|$ is 'Δp', since $(2a+b+c)$ becomes the maximum value that can be included in 'Δp', $(2a+b+c) < x/2$. Accordingly, $(2a+b+c)$ should satisfy the following formula: $0 < 2(2a+b+c) < x$.

In this respect, if the change in the heartbeat is desired to be in a predetermined suitable range, that is, in the range from $(x-\Delta p)$ to $(x+\Delta p)$, the current heartbeat should be within the next heartbeat transmission time. Besides, in consideration of the time required for receiving and processing, the transmission period 'p' to be changed should satisfy the following formula: $x-(2a+b+c) < p < x+(2a+b+c)$.

Accordingly, $-(2a+b+c) < p-x < (2a+b+c)$, that is, $|p-x| < (2a+b+c)$. In this respect, on the basis of the above definition, since $\Delta p = |p-x|$, $\Delta p < (2a+b+c)$.

The above formula can be extended to the following formula: $(2a+b+c) < x - (2a+b+c) < p < 3(2a+b+c) < x + (2a+b+c)$.

Therefore, the maximum tolerance of the period change 'Δp' should be within (2a+b+c), and the transmission period 'p' to which the change is actually applied should be continuously changed in the range from x-(2a+b+c) to x+(2a+b+c).

The random generation process will now be described according to an embodiment based on an experiment.

Values 'a', 'b' and 'c' may be varied depending on the system specification and a network environment. In this case, they can be set by correcting a configuration by tuning when a system is set up.

In the experiment, a value obtained by averaging values simply measured for 10 times on the basis of a TX1A system was used. For the simplicity, we used a configuration file as the way of storing measured values.

Test system: SPARC 10 dual CPU Unix Processor Board

x: the fixed heartbeat period : 500 ms

a: an average transmission time : 14.7 ms

b: an average heartbeat processing time : 1.2 ms

c: an average a state transition time : 2.8 ms

(2a+b+c) = 33.4 ms > |p-x|

In a configuration file,

```
AD.HB.VAR_LIMIT_SEC = 0
AD.HB.VAR_LIMIT_USEC = 33400
AD.HB.PERIOD_SEC = 0
AD.HB.PERIOD_USEC = 500000
...
```

variable seed : a random seed

variable hbVarLimit : a limit that heartbeat period can vary

variable x : the fixed heartbeat period

variable prevrange : the old varied range within the limit

variable newrange : the new acceptable range to be varied at a new period within the limit

variable p : the heartbeat period to be newly changed

*** the time unit of the above all period related variables is micro second.

```
/ * pseudo code for initialization of a system */
concern the unique processor id as the random seed and assign it to a
variable seed ;
   / * seed = processor_id ; * /
initialize the random number generator with the seed value ;
   / * randomize(seed) ; */
read an environment variable VAT_LIMIT_USEC and store it to a variable
hbVarLimit
   / * hbVarLimit = get_parameter(HB.VAR_LIMIT_USEC) ; */
read an environment variable PERIOD_USEC and store it to a variable x ;
   / * x = get_parameter(HB.PERIOD_USEC) ; */
initialize a variable prevrange ;
   / * prevrange = 0 */
...

/ * pseudo code in outgoing heartbeat processing block */
{
```

```
choose a random number in the range of hbVarLimit and assign it to a variablble newrange;
   / * newrange = random() % hbVarLimit + 1; 1~33400 */
determine the sign of the newrange ;
   / * newrange * = (random() % 2 ? 1: -1) ; by dividing the value by 2, if a remainder is 1, it gets +value, while if the remainder is 0, it gets –value */
assign the newly modified period to a variable x ;
   / * p = x –prevrange + newrange ; a previous modified value is corrected to give a change in the fixed period */
cancel the previous timer
   / * cancel_time (outgoingTimer_) ; a previous timer is finished */
schedule a new timer to execute sendHeartbeatFunction block after p micro seconds
   / * outgoingTimer = schedule_time (thisObject, sendHeartbeatFunction, 0, p ; sec = 0, usec = p */
store the newrange as a prevrange
   / * prevrange = newrange ; */
}
```

The heartbeat transmission process of the outgoing heartbeat processing block 13 in which the heartbeat is transmitted by generating the random numbers will now be described with reference to Figure 4.

First, in order to transmit state information applied from the duplication FSM processing block 12 to the twin, the random numbers are generated according to the above-stated process and used to generate the transmission period of a heartbeat (S11).

And then, a timer is scheduled and started(S12). As a transmission time determined at step S11 lapses (S15), a heartbeat carrying self state information is transmitted to the twin (S16), and then it returns to the step S11.

When the outgoing heartbeat processing block 13 transmits the heartbeat through the above process, the incoming heartbeat processing block of the other processor receives the heartbeat and informs its own duplication FSM processing block of the state information of the twin, as shown in Figure 5.

First, the incoming heartbeat processing block schedules and starts the timer (S21) and waits for the heartbeat to be transmitted from the twin for a predetermined time. In this respect, the predetermined time is determined as a sufficient value greater than the maximum value of the heartbeat transmission period.

When the incoming heartbeat processing block receives the heartbeat from the twin within a predetermined time (S22), it stops the timer (S24) and transmits the received state information of the twin to the duplication FSM processing block (S25), and goes back to step S21.

Meanwhile, if no heartbeat is transmitted from the twin until a predetermined time lapses, the incoming heartbeat processing block judges that the twin has been down, and transmits information related to the down to the duplication FSM block (S25).

As so far described, according to the method for controlling duplicated processors of the present invention, when the two processors start, seeds for the random number are differently allocated to generate different random numbers, and the heartbeat transmission period is continuously changed by using the random numbers to differentiate the transmission and receiving time of the heartbeat between the two processors. Therefore, an anomalous dual state transition, that is, a state fluctuation phenomenon that dual ACTIVE and dual STANDBY are repeatedly performed that may occur in concurrently receiving the heartbeat by two processors, can be prevented from occurring.

In addition, at the time when a network-related disturbance is restored, since each transmission period of the two processors is differently changed to each other, so that a prompt restoration can be ensured.

As the present invention may be embodied in several forms without departing from the spirit or essential characteristics thereof, it should also be understood that the above-described embodiments are not limited by any of the details of the foregoing description, unless otherwise specified, but rather should be construed broadly within its spirit and scope as defined in the appended claims, and therefore all changes and modifications that fall within the meets and bounds of the claims, or equivalence of such meets and bounds are therefore intended to be embraced by the appended claims.